

**NAME**

**dwarf\_attrval\_flag**, **dwarf\_attrval\_signed**, **dwarf\_attrval\_string**, **dwarf\_attrval\_unsigned** - retrieve the value of an attribute within a DWARF debugging information entry

**LIBRARY**

DWARF Access Library (libdwarf, -ldwarf)

**SYNOPSIS**

```
#include <libdwarf.h>
```

*int*

```
dwarf_attrval_flag(Dwarf_Die die, Dwarf_Half attr, Dwarf_Bool *ret, Dwarf_Error *err);
```

*int*

```
dwarf_attrval_signed(Dwarf_Die die, Dwarf_Half attr, Dwarf_Signed *ret, Dwarf_Error *err);
```

*int*

```
dwarf_attrval_string(Dwarf_Die die, Dwarf_Half attr, const char **ret, Dwarf_Error *err);
```

*int*

```
dwarf_attrval_unsigned(Dwarf_Die die, Dwarf_Half attr, Dwarf_Unsigned *ret, Dwarf_Error *err);
```

**DESCRIPTION**

These functions search the debugging information entry referenced by argument *die* for the attribute named by argument *attr*. If the named attribute is found, the functions set the location pointed to by argument *ret* to the value of the attribute. The argument *err*, if non NULL, will be used to return an error descriptor in case of an error.

Function **dwarf\_attrval\_flag**() sets the location pointed to by argument *ret* to either 0 or 1. If the form of the attribute named by argument *attr* is DW\_FORM\_flag, function **dwarf\_attrval\_flag**() sets the location pointed to by argument *ret* to 1 if the attribute has a non-zero value, or to 0 otherwise. If the form of the attribute named by argument *attr* is DW\_FORM\_flag\_present, function **dwarf\_attrval\_flag**() unconditionally sets the location pointed to by argument *ret* to 1. The form of the attribute must be one of DW\_FORM\_flag or DW\_FORM\_flag\_present.

Function **dwarf\_attrval\_signed**() stores the value for the attribute named by argument *attr*, into the location pointed to by argument *ret*. The attribute's value is treated as a signed integral quantity and is sign-extended as needed. The attribute named by the argument *attr* must belong to the CONSTANT class and must have one of the following forms: DW\_FORM\_data1, DW\_FORM\_data2, DW\_FORM\_data4, DW\_FORM\_data8 or DW\_FORM\_sdata.

Function **dwarf\_attrval\_string()** sets the location pointed to by argument *ret* to a pointer to a NUL-terminated string that is the value of the attribute named by argument *attr*. The form of the attribute must be one of DW\_FORM\_string or DW\_FORM\_strp.

Function **dwarf\_attrval\_unsigned()** stores the value for the attribute named by argument *attr* into the location pointed to by argument *ret*. The attribute's value is treated as an unsigned integral quantity, and is zero-extended as needed. The named attribute must belong to one of the CONSTANT, ADDRESS or REFERENCE classes and must have one of the following forms: DW\_FORM\_addr, DW\_FORM\_data1, DW\_FORM\_data2, DW\_FORM\_data4, DW\_FORM\_data8, DW\_FORM\_uda, DW\_FORM\_ref1, DW\_FORM\_ref2, DW\_FORM\_ref4, DW\_FORM\_ref8, or DW\_FORM\_ref\_uda.

If the attribute named by argument *attr* is DW\_AT\_type and is not present in the debugging information entry referenced by argument *die*, and if a DW\_AT\_abstract\_origin or DW\_AT\_specification attribute is present in the debugging information entry, function **dwarf\_attrval\_unsigned()** will search for the named attribute in the debugging information entry referenced by the DW\_AT\_abstract\_origin or DW\_AT\_specification attribute.

## RETURN VALUES

On success, these functions return DW\_DLV\_OK. If the named attribute was not found in the specified debugging information entry descriptor these functions return DW\_DLV\_NO\_ENTRY and set argument *err*. For other errors, these functions return DW\_DLV\_ERROR and set argument *err*.

## COMPATIBILITY

These functions are extensions added by this implementation of the DWARF(3) API.

## ERRORS

These functions may fail with the following errors:

[DW_DLE_ARGUMENT]	Either of the arguments <i>die</i> or <i>ret</i> was NULL.
[DW_DLE_NO_ENTRY]	Argument <i>die</i> did not contain an attribute corresponding to the value in argument <i>attr</i> .
[DW_DLE_ATTR_FORM_BAD]	The attribute named by argument <i>attr</i> was not of a permitted form.

## SEE ALSO

dwarf(3), dwarf\_attr(3), dwarf\_hasattr(3)